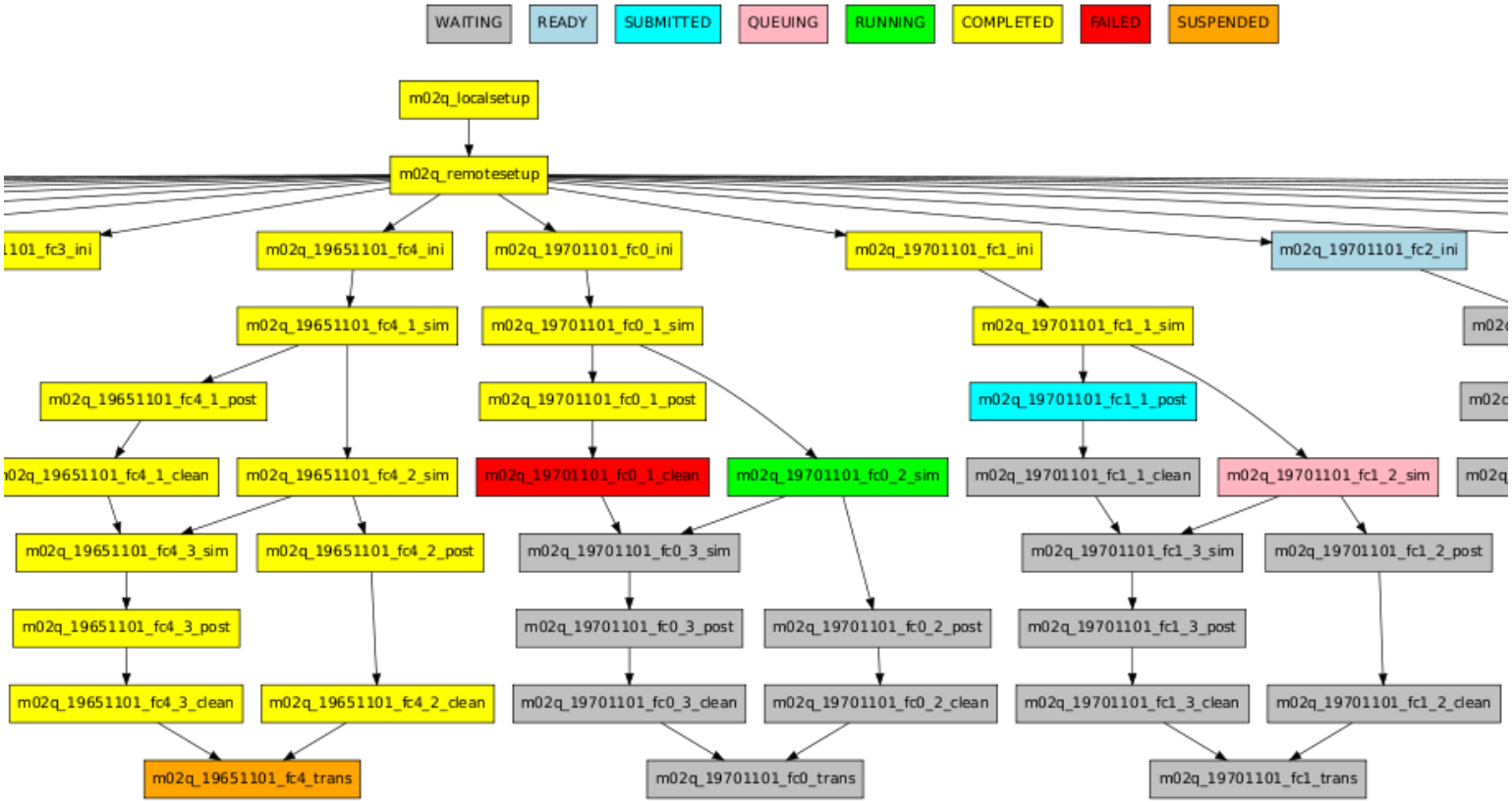
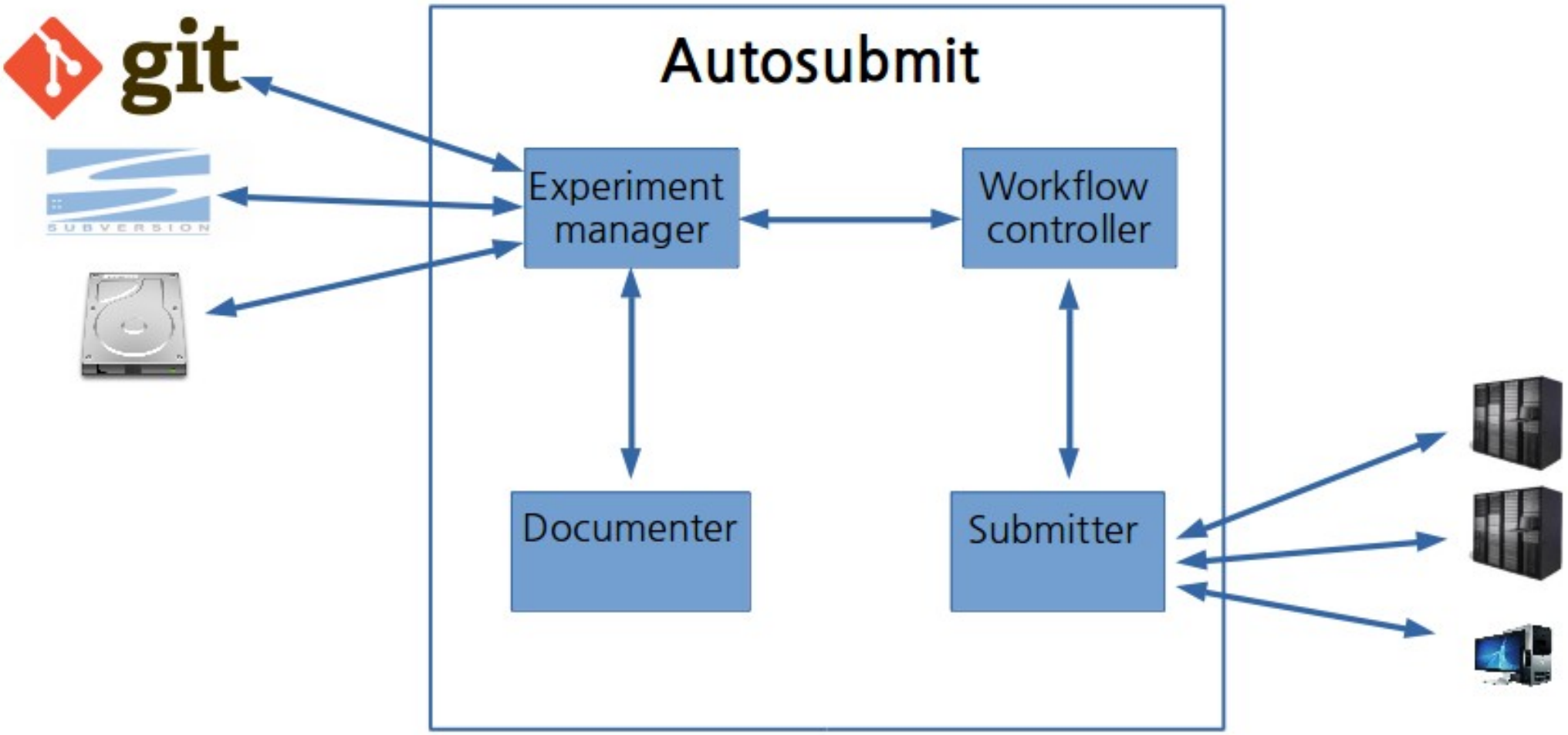
Autosubmit 3.1



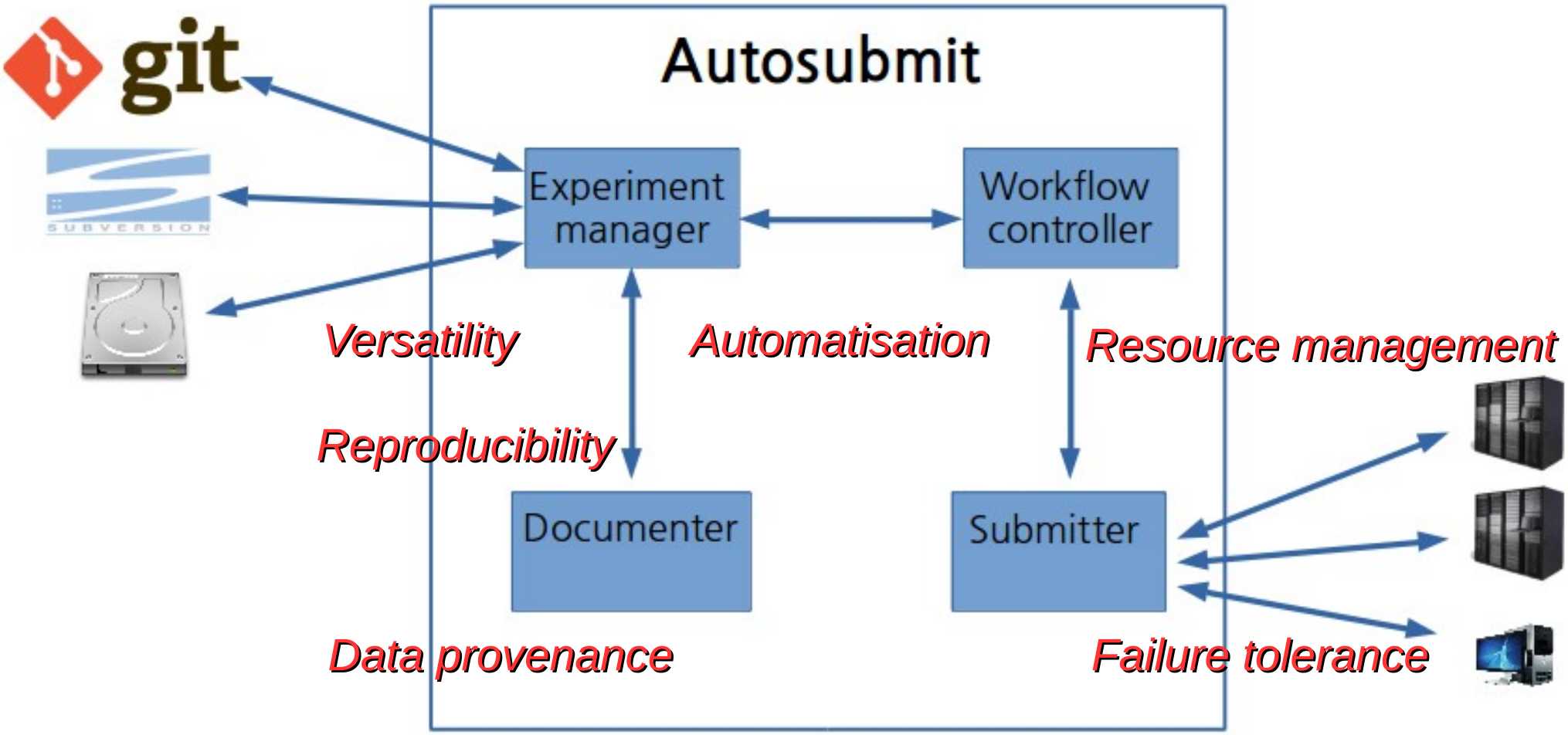
D. Manubens, J. Vegas

O. Mula-Valls, M. Asif, K. Serradell, PA. Bretonnière, V. Guemas, C. Prodhomme, O. Bellprat, F. J. Doblas-Reyes

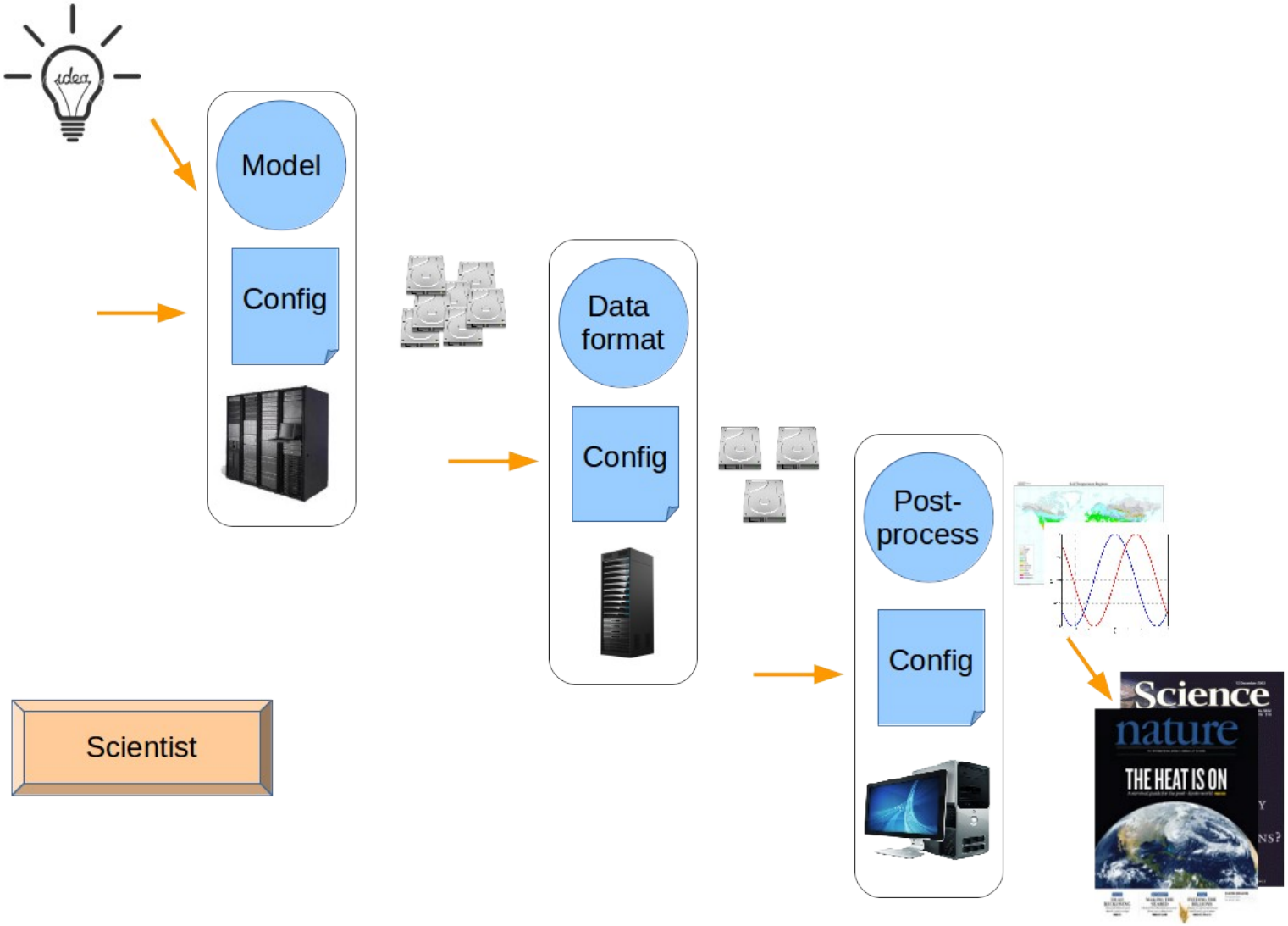
What is Autosubmit ?



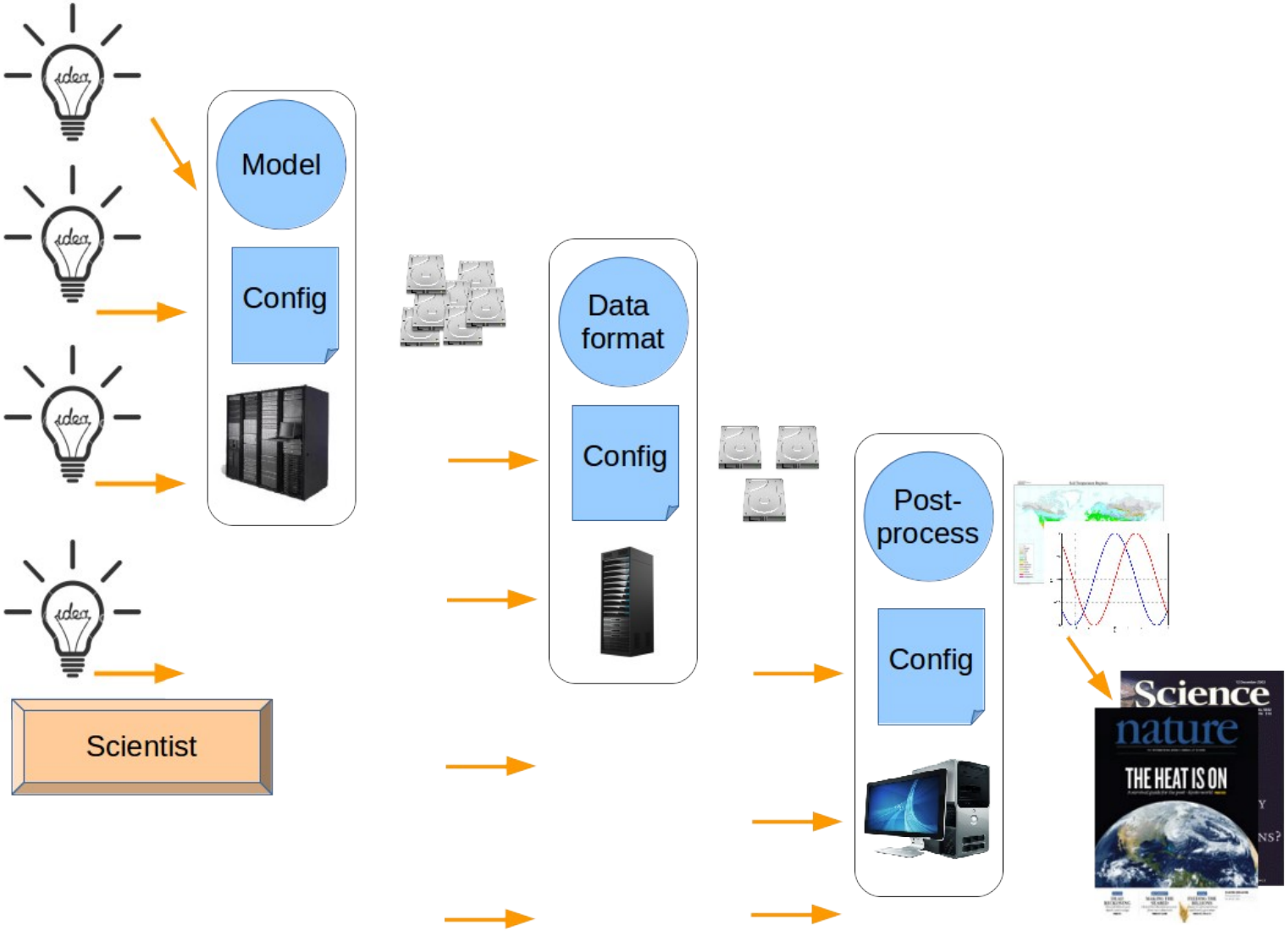
Why is Autosubmit needed ?



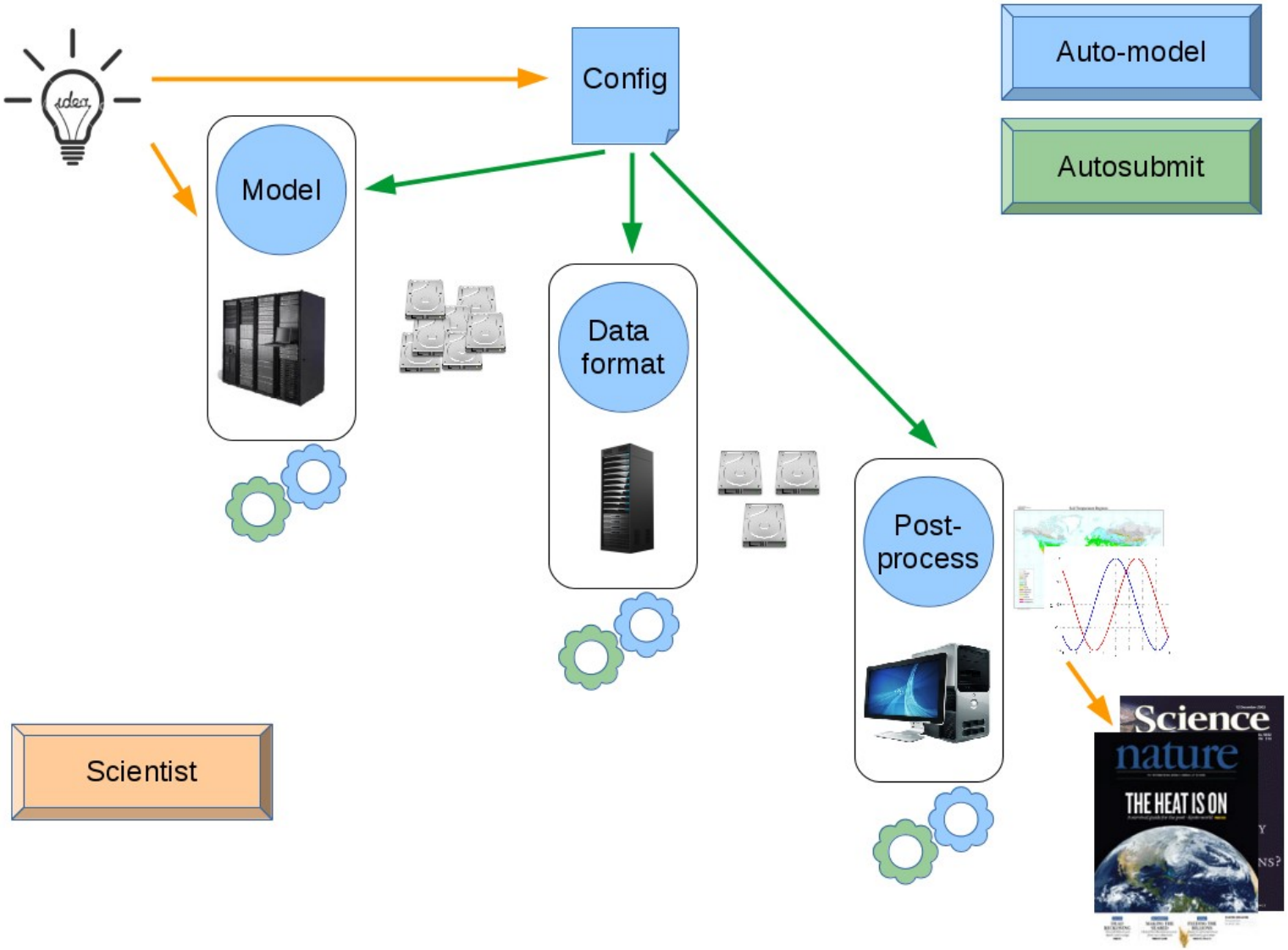
Why is Autosubmit needed ?



Why is Autosubmit needed ?



Why is Autosubmit needed ?

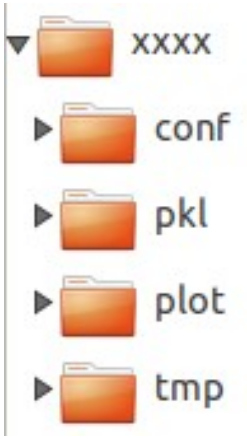


How does Autosubmit work ?

Python Package Index (PyPI)

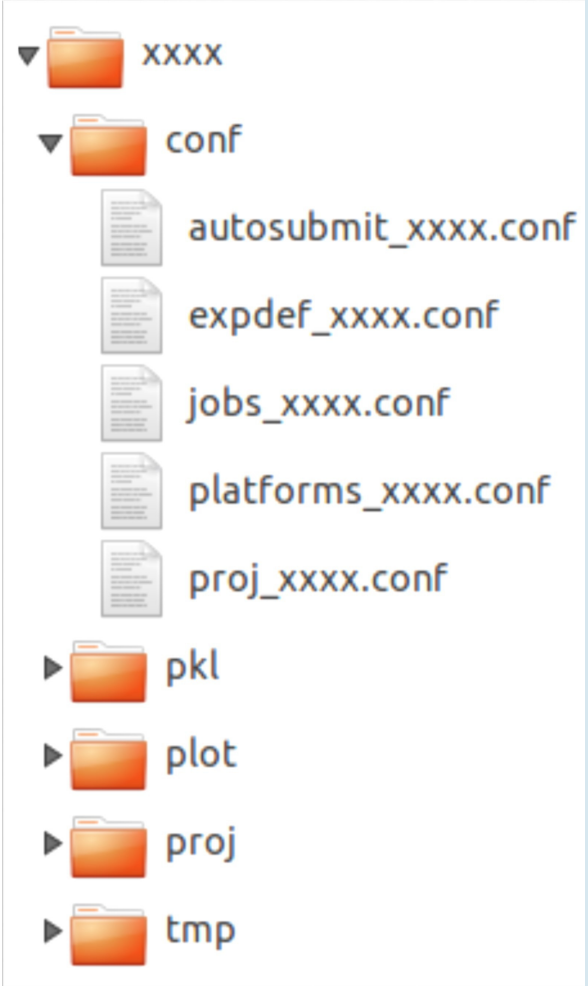
Experiment creation

```
autosubmit expid –H HPCname
```



Experiment creation

```
autosubmit expid –H HPCname
```



Experiment configuration

```
autosubmit create xxxx
```

Start dates, members and chunks (number and length).

Experiment project source: origin (version control system or path) and project configuration file path.

expdef_xxxx.conf

Workflow to be run: scripts to execute, dependencies between tasks, task requirements (processors, wallclock time...) and platform to use.

jobs_xxxx.conf

HPC, fat-nodes and supporting computers configuration.

Usually provided by technicians, users will only have to change login and accounting options for HPCs.

platforms_xxxx.conf

Project dependant experiment variables that Autosubmit will substitute in the scripts to be run.

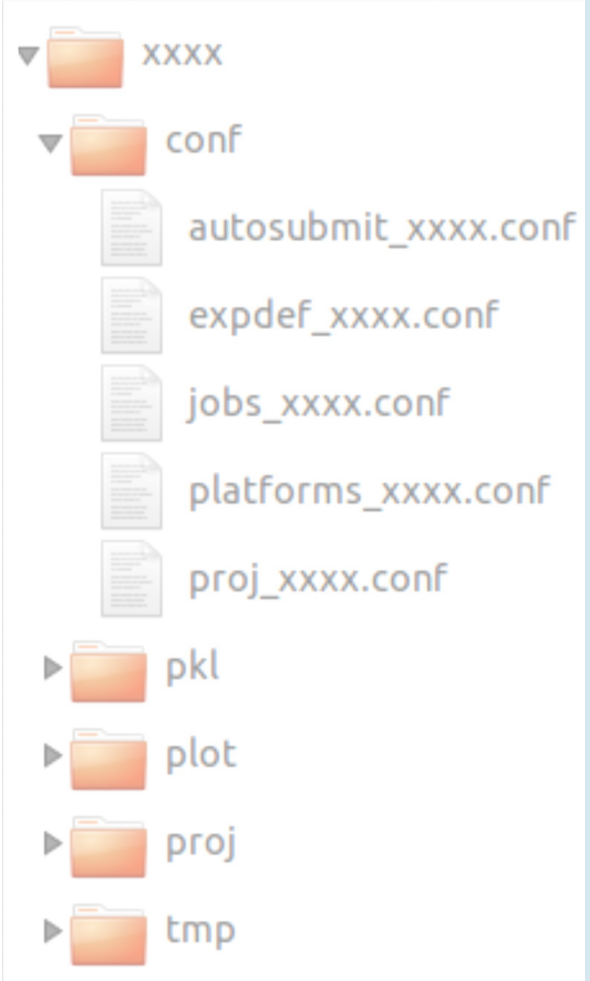
proj_xxxx.conf

Experiment creation

```
autosubmit expid –H HPCname
```

Experiment configuration

```
autosubmit create xxxx
```



Start dates, members and chunks (number and length).

Experiment project source: origin (version control system or path) and project configuration file path.

expdef_xxxx.conf

Workflow to be run: scripts to execute, dependencies between tasks, task requirements (processors, wallclock time...) and platform to use.

jobs_xxxx.conf

HPC, fat-nodes and supporting computers configuration.

Usually provided by technicians, users will only have to change login and accounting options for HPCs.

platforms_xxxx.conf

Project dependant experiment variables that Autosubmit will substitute in the scripts to be run.

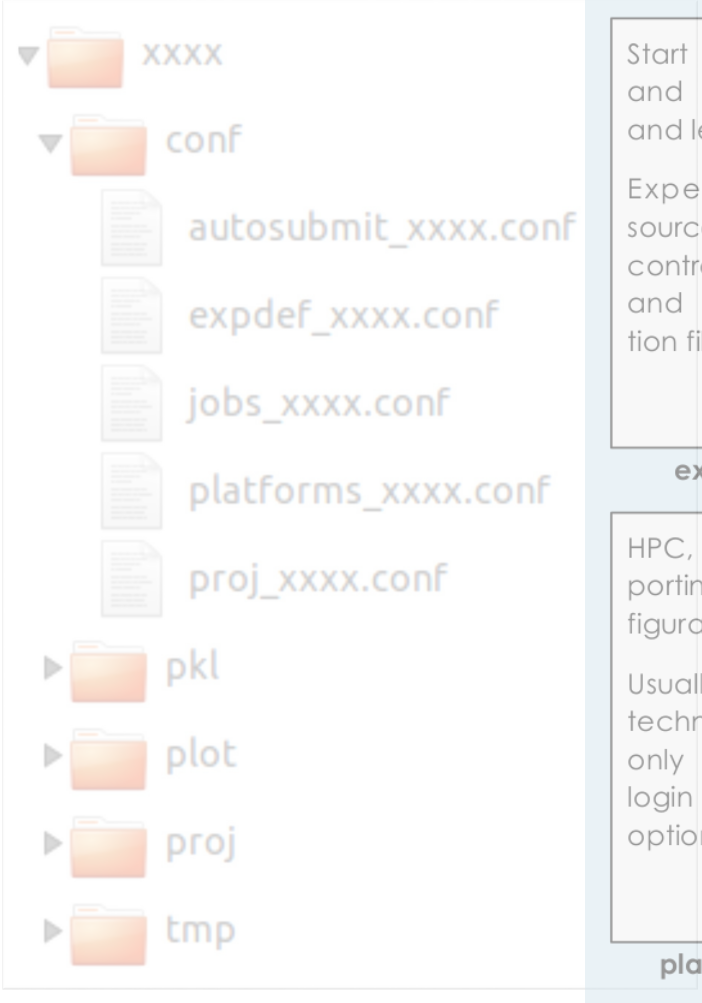
proj_xxxx.conf

Experiment configuration file

```
[project]
# Select project type. STRING = git, svn, local, none
# If PROJECT_TYPE is set to none, Autosubmit self-contained dummy templates will be used
PROJECT_TYPE = git
# Destination folder name for project. type = STRING, default = leave empty,
PROJECT_DESTINATION = model

# If PROJECT_TYPE is not git, no need to change
[git]
# Repository URL  STRING = 'https://github.com/torvalds/linux.git'
PROJECT_ORIGIN = https://gitlab.cfu.local/cfu/auto-ecearth3.git
# Select branch or tag, STRING, default = 'master',
# help = {'master' (default), 'develop', 'v3.1b', ...}
PROJECT_BRANCH = develop
# type = STRING, default = leave empty, help = if model branch is a TAG leave empty
PROJECT_COMMIT =

# If PROJECT_TYPE is not svn, no need to change
[svn]
# type = STRING, help = 'https://svn.ec-earth.org/ecearth3'
PROJECT_URL =
# Select revision number. NUMERIC = 1778
PROJECT_REVISION =

# If PROJECT_TYPE is not local, no need to change
[local]
# type = STRING, help = /foo/bar/ecearth
PROJECT_PATH =

# If PROJECT_TYPE is none, no need to change
[project_files]
# Where is PROJECT CONFIGURATION file location relative to project root path
FILE_PROJECT_CONF = templates/ecearth3/ecearth3.conf
```

Experiment creation

```
autosubmit expid –H HPCname
```



Experiment configuration

```
autosubmit create xxxx
```

Start dates, members and chunks (number and length).

Experiment project source: origin (version control system or path) and project configuration file path.

expdef_xxxx.conf

Workflow to be run: scripts to execute, dependencies between tasks, task requirements (processors, wallclock time...) and platform to use.

jobs_xxxx.conf

HPC, fat-nodes and supporting computers configuration.

Usually provided by technicians, users will only have to change login and accounting options for HPCs.

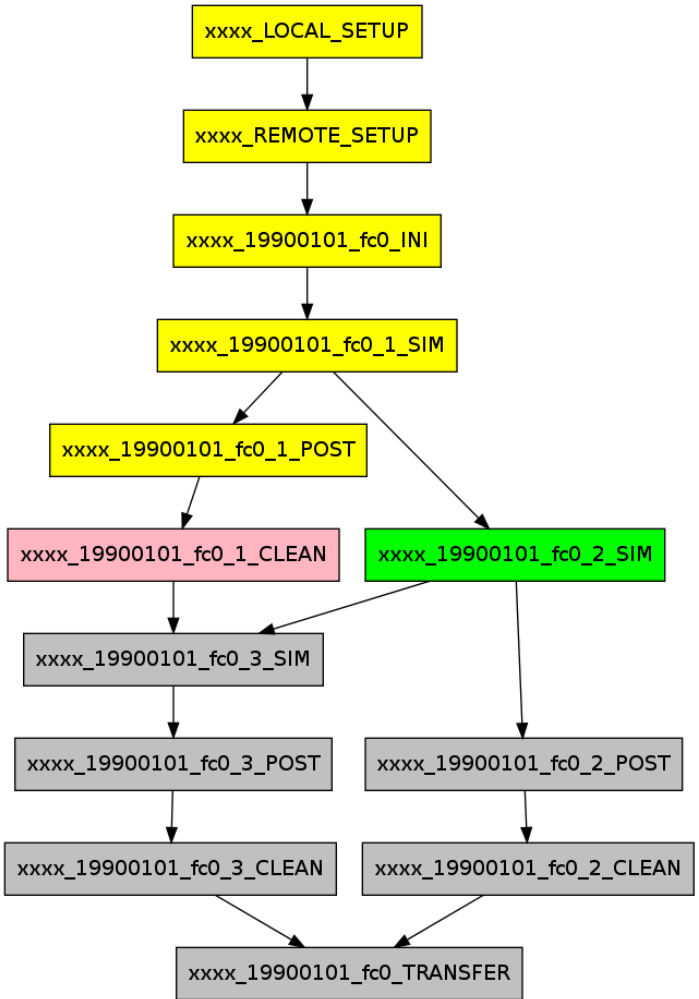
platforms_xxxx.conf

Project dependant experiment variables that Autosubmit will substitute in the scripts to be run.

proj_xxxx.conf

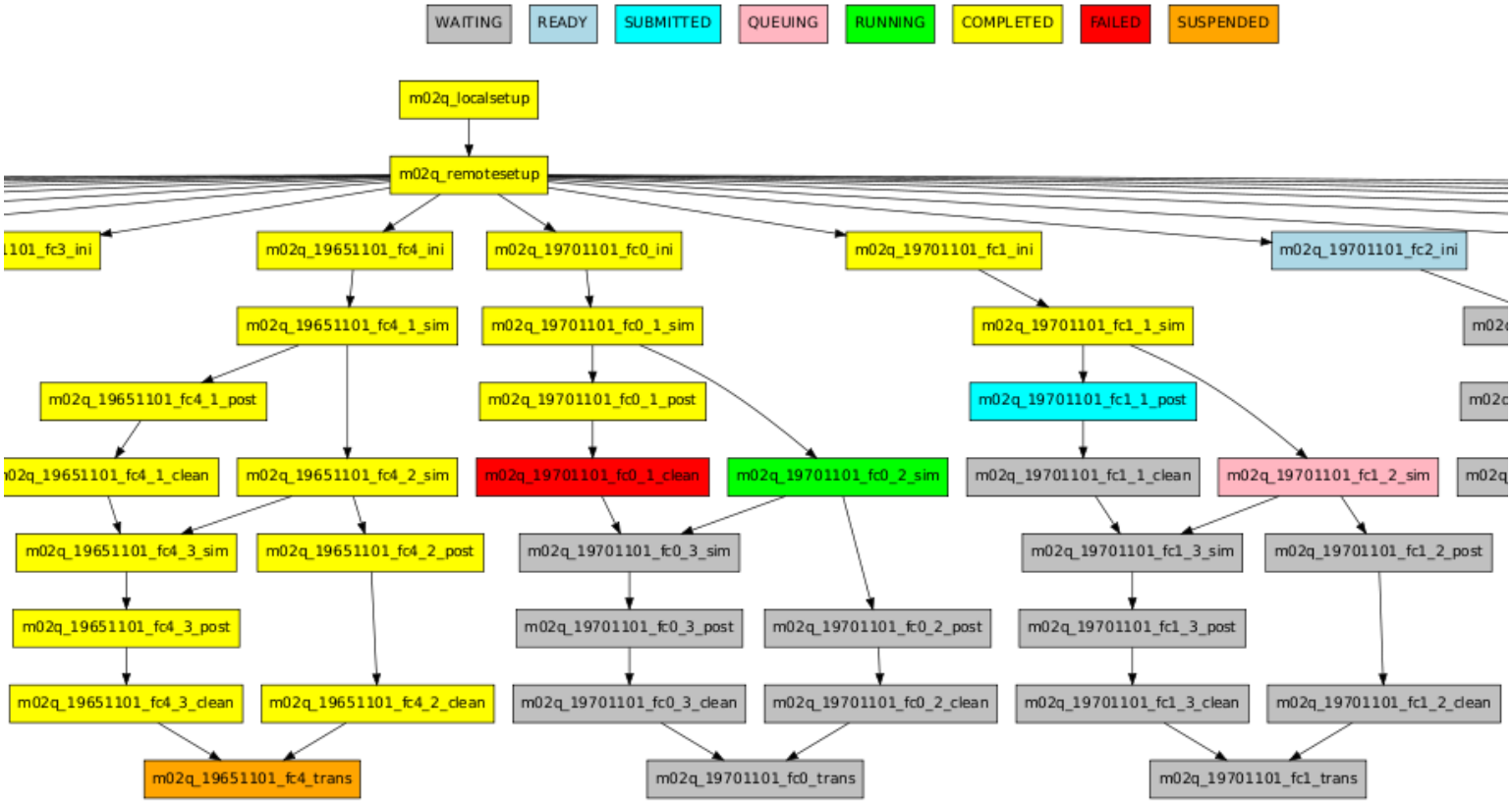
Experiment run

```
autosubmit run xxxx
```



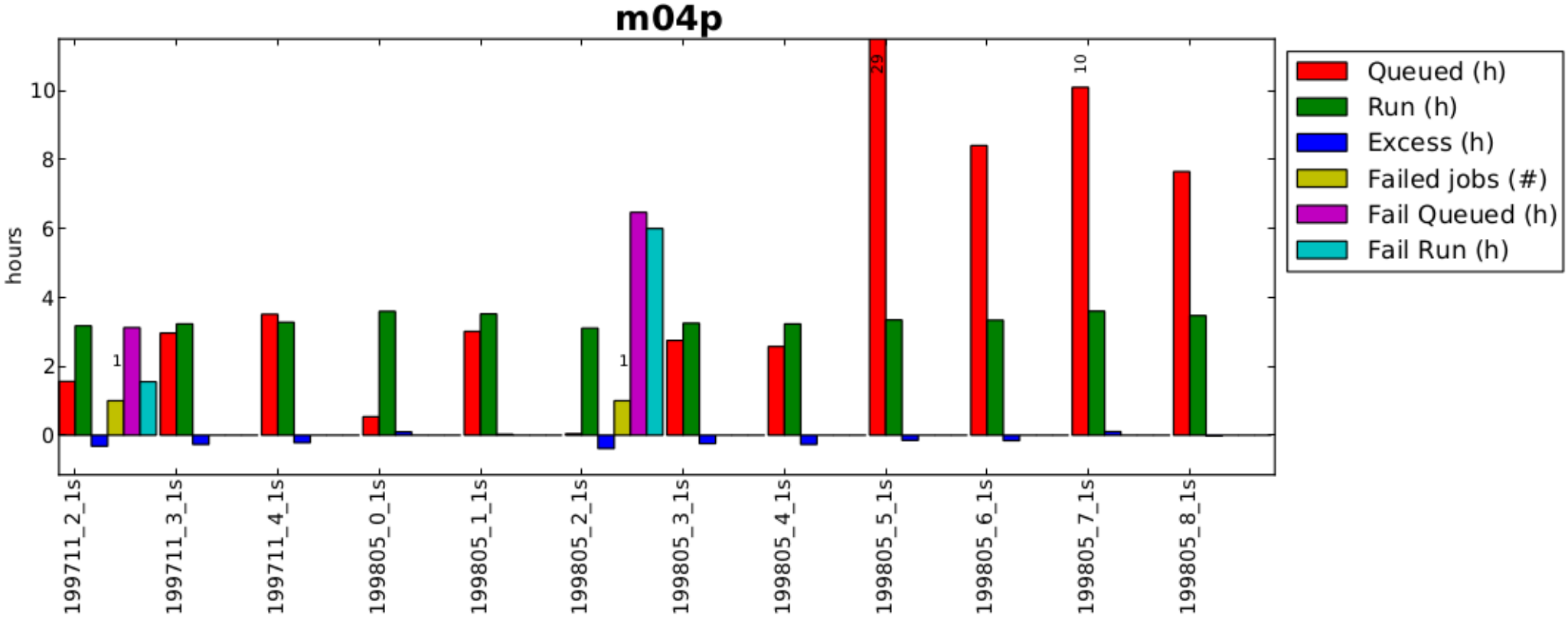
Experiment monitoring

```
autosubmit monitor xxxx
```

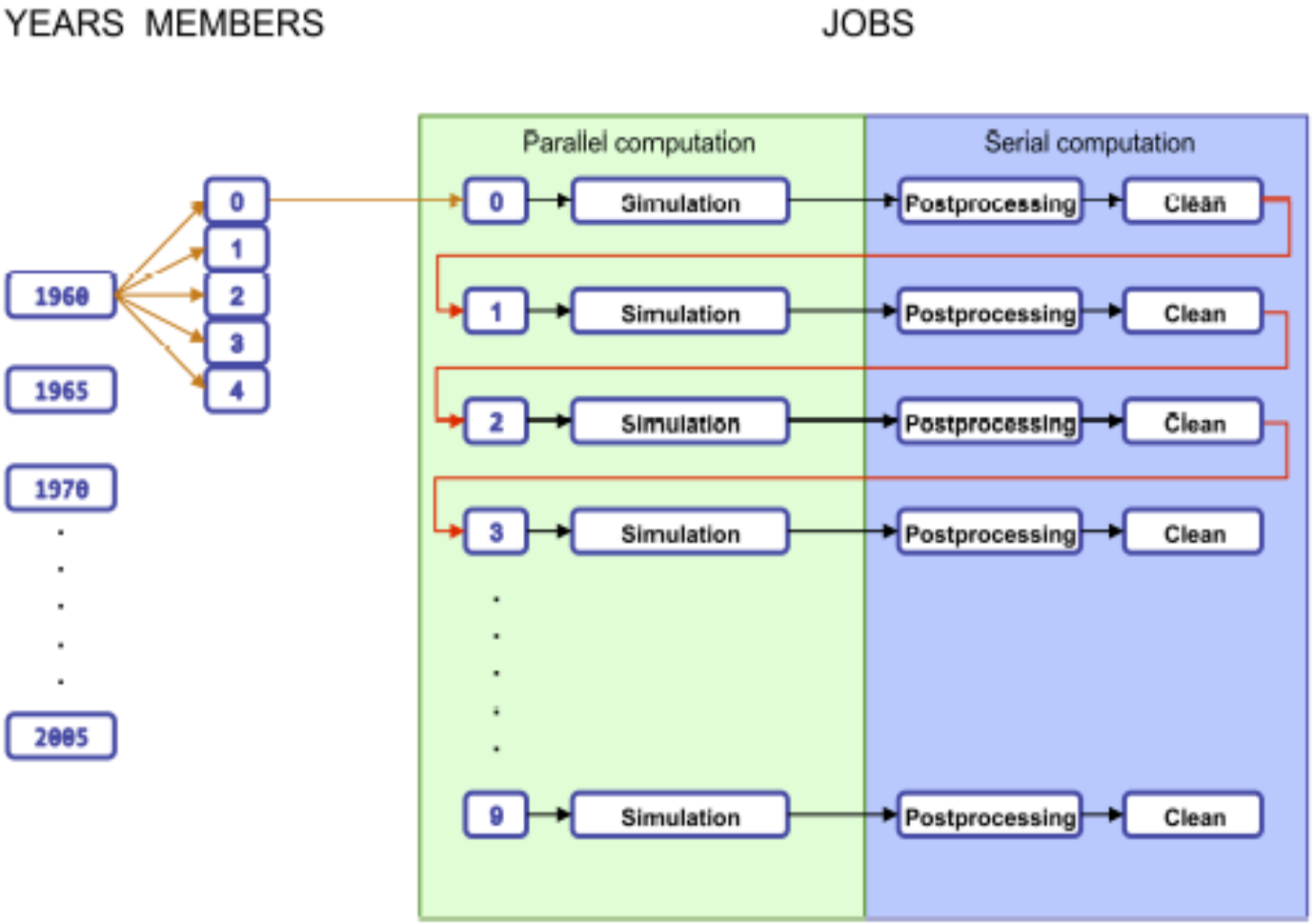


Automatic statistics

```
autosubmit stats cxxx
```

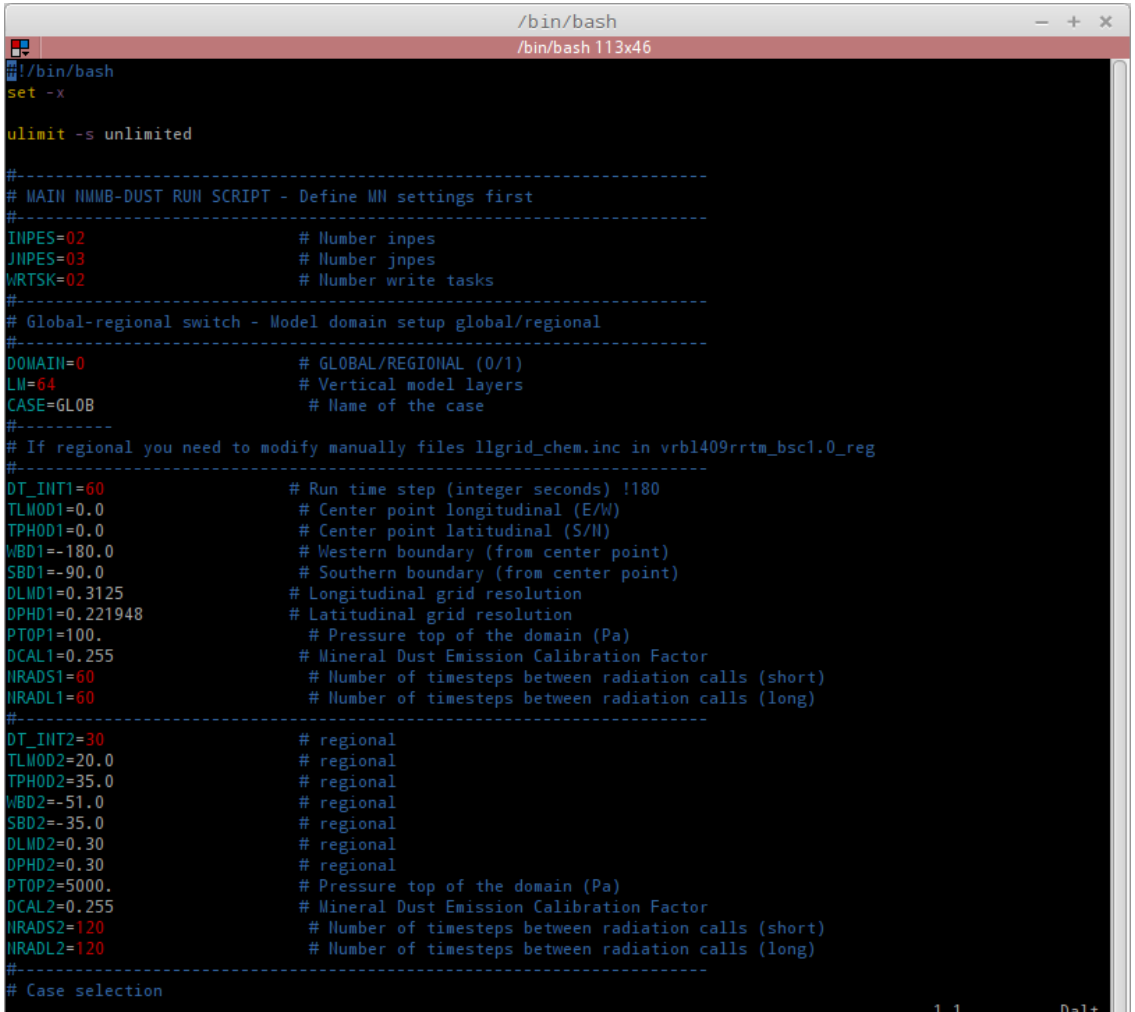


EC-Earth/IC3 workflow



NMMB/BSC workflow

- Currently ~ 1.000 lines of code
- Splited by blocks (fixed, variable, model…)



Splitting tasks

- We divide the script and we create five different scripts
  - FIX
  - VAR
  - SIM
  - POST

- And we define dependences

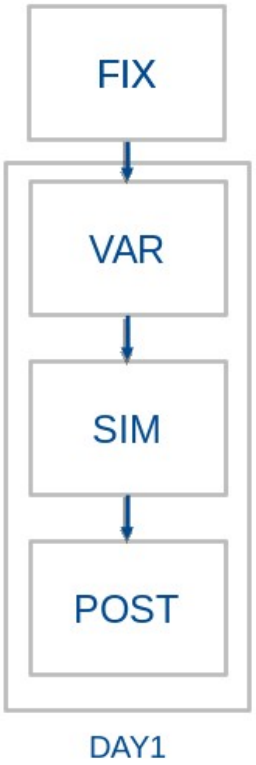
Splitting tasks

- We divide the script and we create five different scripts
  - FIX
  - VAR
  - SIM
  - POST

- And we define dependences

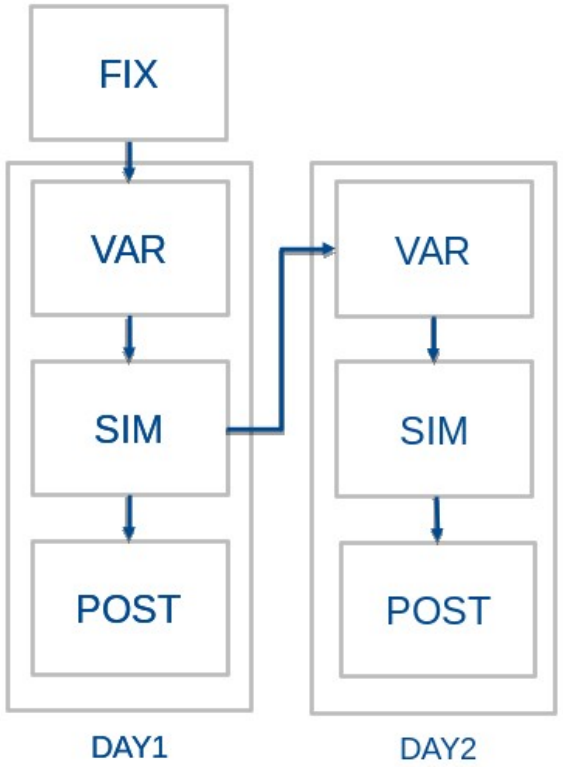
Splitting tasks

- We divide the script and we create five different scripts
  - FIX
  - VAR
  - SIM
  - POST
- And we define dependences



Splitting tasks

- We divide the script and we create five different scripts
  - FIX
  - VAR
  - SIM
  - POST
- And we define dependences



Future Work

- Create a wrapper for jobs that allows to send various independent tasks as one big job to reduce queueing time and to allow participation in tier-0 projects.

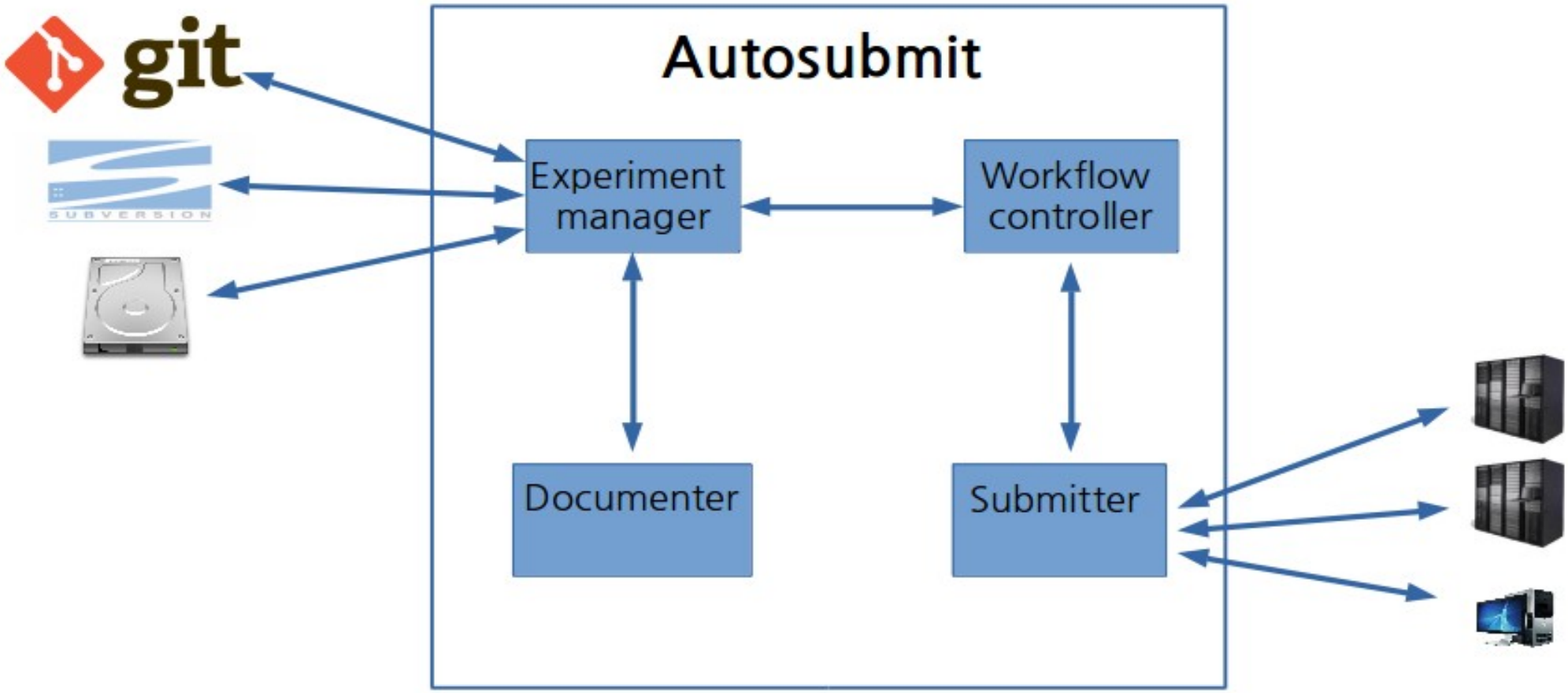
- Implement the SAGA standard for job submission & control.

- Expand data stored in database to store all data needed for CMIP6. Use the es-doc python API.

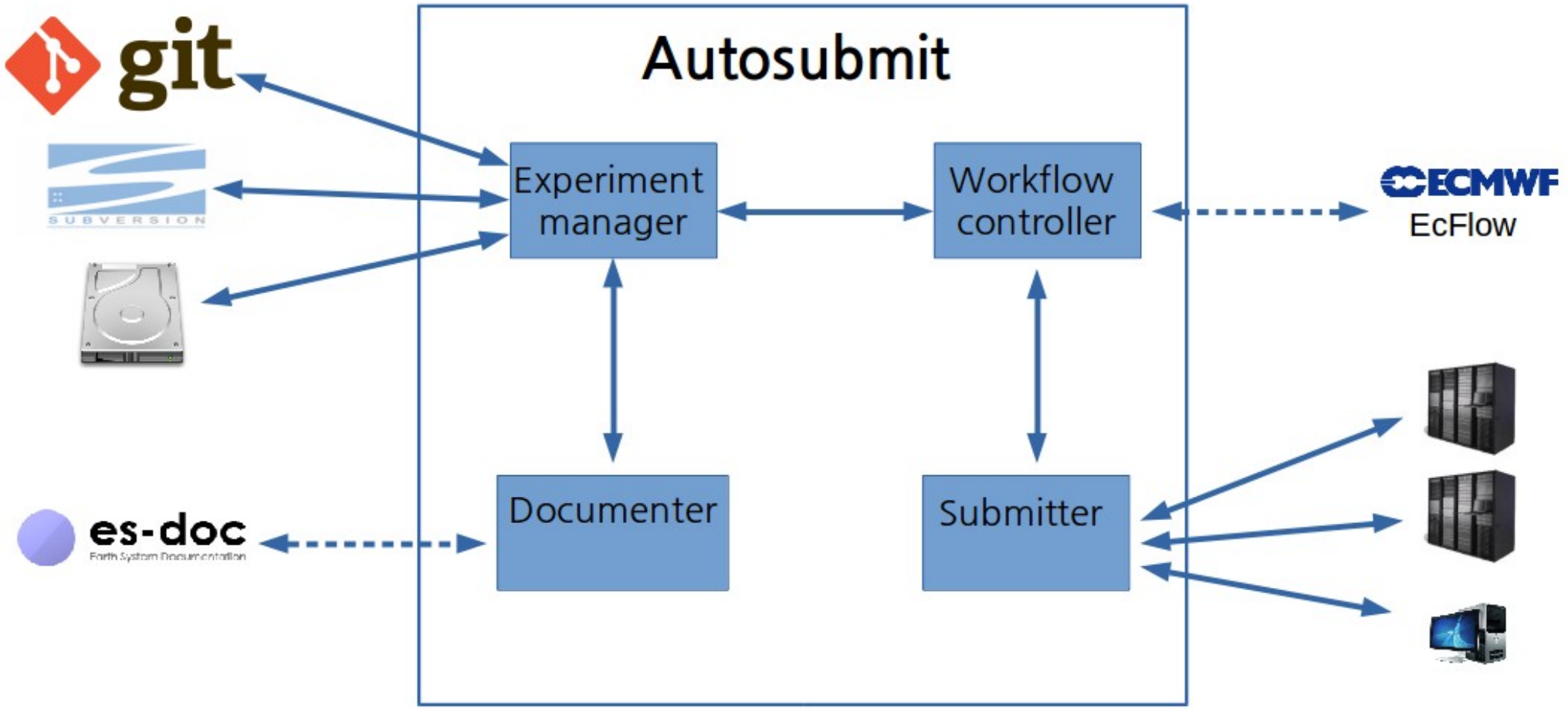
- Build a plugin system for different models databases.

- Evaluate the suitability of using ecFlow (https://software.ecmwf.int/wiki/display/ECFLOW) and Cylc (http://cylc.github.io/cylc) for workflow configuration to converge with community standards.

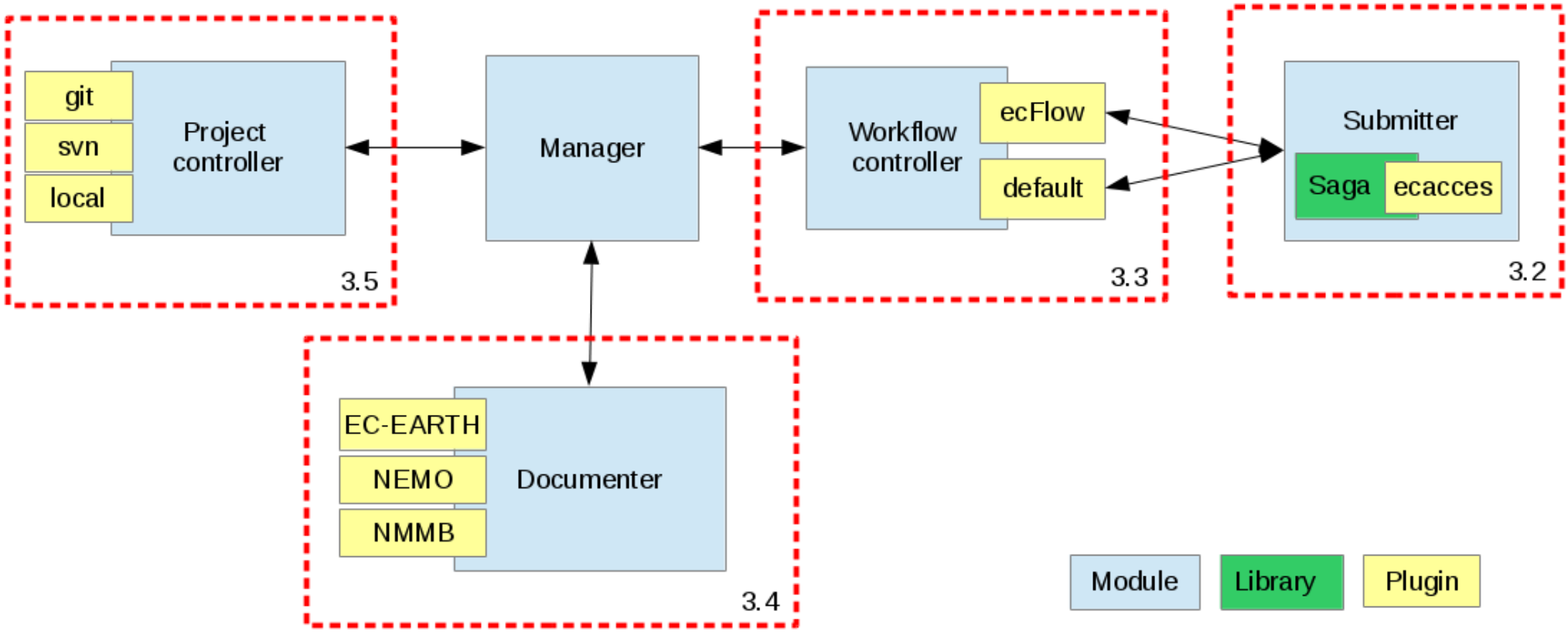
Future Work


git
SUBVERSION
Autosubmit
Experiment manager
Workflow controller
Documenter
Submitter

Future Work


git
SUBVERSION
Autosubmit
Experiment manager
Workflow controller
ECMWF
EcFlow
es-doc
Documenter
Submitter

Roadmap


git
svn
local
Project controller
3.5
Manager
Workflow controller
ecFlow
default
3.3
Submitter
Saga
ecacces
3.2
EC-EARTH
NEMO
NMMB
Documenter
3.4
Module
Library
Plugin